

TEKNIIKAN JA LIIKENTEEN TOIMIALA

Tietotekniikka

Ohjelmistotekniikka

INSINÖÖRITYÖ

Jäsentietokantaohjelmiston kehittäminen

Työn tekijä: Mikko Kulmala
Työn valvoja: Erja Nikunen
Työn ohjaaja: Anni Reissell

Työ hyväksytty: 16. 4. 2007

Erja Nikunen
Yliopettaja

ALKULAUSE

Tämä työ on Helsingin ammattikorkeakoulun ohjelmistotekniikan päättötyö. Työ on tehty lukuvuosien 2006 - 2007 aikana.

Työni valvojana toimi Helsingin ammattikorkeakoulun puolesta yliopettaja Erja Nikunen, jota kiitän ymmärtäväisestä suhtautumisesta, kärsivällisestä seurannasta ja ohjaamisesta.

Haluan myös kiittää erityisesti vaimoani, joka on ollut koko projektin ajan erittäin ymmärtäväinen ja kannustava.

Lisäksi kiitän kaikkia muita henkilöitä, jotka ovat osaltaan myötävaikuttaneet insinööri-työni edistymiseen ja onnistumiseen.

Helsingissä 16.4.2007

Mikko Kulmala

INSINÖÖRITYÖN TIIVISTELMÄ

Tekijä: Mikko Kulmala	
Työn nimi: Jäsentietokantaohjelmiston kehittäminen	
Päivämäärä: 16.4.2007	Sivumäärä: 31 s.
Koulutusohjelma: Tietotekniikka	Suuntautumisvaihtoehto: Ohjelmistotekniikka
Työn valvoja: Erja Nikunen	
Työn ohjaaja: Anni Reissell	
<p>iLEAPS (Integrated Land Ecosystem-Atmosphere Processes Study) on kansainvälinen tutkimusohjelma, jonka tavoitteena on parantaa maa-ilmakehä rajapinnan tuntemusta.</p> <p>Opinnäytetyönä on tehty iLEAPS:ille “jäsenrekisteri”-sovellus, jota on myöhemmin muokattu parantamalla muunmuassa käyttöliittymää. Lisäksi ohjelman eri palaset koottiin yhteen moduuleiksi, jonne on kerätty yhteen kuuluvia asioita kuten funktiot ja uuteen versioon tehdyt luokat. Sovellus on toteutettu PHP/MySQL-työkaluilla. Ohjelma koostuu kahdesta osasta: julkisilta www-sivuilta löytyvästä lomakkeesta sekä varsinaisesta tietokannan hallintatyökalusta.</p> <p>Lomakkeella pystyy ilmoittamaan itsensä halukkaaksi liittymään mukaan postituslistalle ja antamaan iLEAPS:ille tarvittavat tiedot itsestään. Hallintatyökalulla voidaan lisätä, poistaa ja muokata henkilöiden tietoja. Työkalun avulla voidaan myös lähettää halutuille ihmisille sähköpostia sekä tulostaa tarvittavia tietoja.</p>	
Avainsanat: PHP, MySQL	

ABSTRACT

Name: Mikko Kulmala	
Title: Jäsentietokantaohjelmiston kehittäminen	
Date: 16.4.2007	Number of pages: 31
Department: Information Technology Study Programme: Software Engineering	
Instructor: Erja Nikunen Supervisor: Anni Reissell	
<p>iLEAPS (Integrated Land Ecosystem-Atmosphere Processes Study) is international research project which scientific goal provide understanding how interacting physical, chemical and biological processes transport and transform energy and matter through the land-atmosphere interface.</p> <p>This program is a memberregister to iLEAPS which has been later modified for example improvements to user interface. This program has been made by PHP/MySQL tools. It includes two part. public www-pages and database administration tool. In public pages you can give your information to iLEAPS. You an use administration tool to add/delete or update information in database.</p>	
Keywords:PHP, MySQL	

SISÄLLYS

ALKULAUSE

TIIVISTELMÄ

ABSTRACT

1 JOHDANTO.....	1
2 TEKNIKOIDEN ESITTELY.....	2
2.1 MySQL.....	2
2.2 Ohjelman tietokannan rakenne.....	3
2.3 PHP.....	5
2.4 Olio-ohjelmointi.....	8
2.5 Siirtyminen funktioista olioihin.....	9
3 TEKNINEN MUUTOS.....	10
3.1 Alkuperäinen versio.....	10
3.2 Päivitetty versio.....	13
3.3 Sovelluksen rakenne.....	15
4 KÄYTTÄJÄN NÄKEMÄT ASIAT.....	20
4.1 Hallinnointityökalu.....	20
4.2 Ilmoittautumislomake.....	25
5 PÄIVITYKSEN HYVÄT JA HUONOT PIIRTEET.....	27
5.1 Ohjelma.....	27
5.2 Moduleihin siirtymisen hyvät ja huonot piirteet.....	27
6 YHTEENVETO.....	29
VIITELUETTELO.....	31

1 JOHDANTO

iLEAPS (Integrated Land Ecosystem-Atmosphere Processes Study) on kansainvälinen tutkimusohjelma, jonka tavoitteena on parantaa maailmakehän rajapinnan tuntemusta.

Opinnäytetyönä on tehty iLEAPS:ille "jäsenrekisterisovellus", jota on myöhemmin muokattu oliopohjaiseksi sekä käyttöliittymää on paranneltu. Sovellus on toteutettu PHP/MySQL-työkaluilla. Ohjelma koostuu kahdesta osasta: julkisilta www-sivuilta löytyvästä lomakkeesta sekä varsinaisesta tietokannan hallintatyökalusta.

Lomakkeella pystyy ilmoittamaan itsensä halukkaaksi liittymään mukaan postituslistalle ja antamaan iLEAPS:ille tarvittavat tiedot itsestään. Hallintatyökalulla voidaan lisätä, poistaa ja muokata henkilöiden tietoja sekä lähettää halutuille ihmisille sähköpostia. Työkalun avulla voidaan myös tulostaa tarvittavia tietoja.

Tavoitteena on ollut saada aikaiseksi helppokäyttöinen ohjelma, jolla pystytään hallinnoimaan helposti iLEAPS:ille kuuluvia projekteja ja projekteihin liittyviä ihmisiä. Tämän takia ohjelman hakutoimintoihinkin on liitetty mahdollisuus hakea juuri tiettyjä asioita tutkivia ihmisiä, jotta tiedetään keneen otetaan yhteyttä, kun jokin asia koskee vain jotakin tiettyä kysymystä.

2 TEKNIKOIDEN ESITTELY

Ohjelma on tehty käyttäen PHP:ta ja se toimii yhdessä MySQL-tietokannan kanssa. Tietokanta ja PHP toimivat molemmat apache www-palvelimella.

Tietokantasovellus tarvitsee toimiakseen tietokannan sekä palvelimen. Tietokantana toimii MySQL (Vers. 11.18) ja palvelimena toimii Linux (Fedora core 3, vers. 2.4.22-1.2199), jossa on päällä apache www-palvelin. Tietokanta ja www-palvelin toimivat samassa koneessa.

Tietokanta (database) voidaan määritellä jotakin käyttötarkoitusta varten laadituksi kokoelmaksi toisiinsa liittyviä säilytettäviä tietoja. Tämän kuvauksen mukaan kuitenkin mikä tahansa säilytettävä kokoelma tietoja olisi tietokanta. Tietokantaan liitetään useita lisävaatimuksia ja ominaisuuksia, kuten tietojen samanaikainen käyttö, tietojen suojaus sekä monipuoliset tiedonhakumahdollisuudet.

Tietokantajärjestelmä muodostuu tietokannasta ja tietokantaa käyttävästä sovelluksesta. Sovellusohjelmat voivat kommunikoida tietokannanhallintajärjestelmän kanssa joko suoraan tai tietoliikenneohjelmiston välityksellä. Samoin tietokannanhallintajärjestelmä saattaa sisäisesti käyttää hyödykseen tietoliikenneohjelmiston palveluita.

2.1 MySQL

MySQL on suosittu ja tehokas SQL-tietokannan hallintajärjestelmä. Se perustuu avoimeen lähdekoodiin. MySQL-tietokannan päälle rakennettu ohjelmointilogiikka tehdään usein PHP-, Python- tai Perl-ohjelmointikielellä, josta tässä työssä kiinnostaa sovelluksessa käytettävä PHP (tästä enemmän luvussa 2.3). Sivut julkaistaan Apache web -palvelimella, joka edelleen toimii Linux-käyttöjärjestelmän päällä.

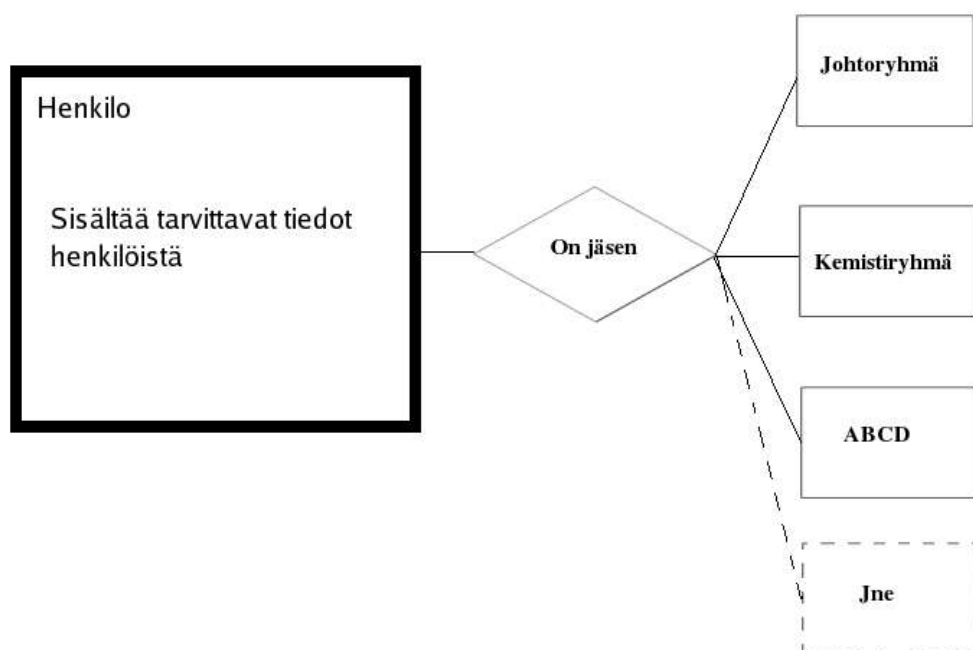
SQL on standardoitu kieli relaatiotietokantojen käsittelyyn ja määrittelyyn. Harri Laineen mukaan [1] SQL:llä voidaan

- määritellä tietokannan käyttäjät ja heidän käyttöoikeutensa
- määritellä tietokannan sisältö
- hakea tietoa tietokannasta, näytölle tai tiedostoon, sovellusohjelman käyttöön
- tehdä päivityksiä tietokantaan vuorovaikutteisesti tai sovellusohjelman kautta
- määritellä talletusrakenteita
- kontrolloida tietokannan samanaikaista käyttöä.

SQL-kieli syntyi alunperin IBM:n San Josen tutkimuslaboratoriossa rakennettaessa relaatiotietokannan prototyyppiä nimeltä System R vuosina 1972–73. SQL:n standardoinnin käynnisti amerikkalainen standardointijärjestö ANSI, joka hyväksyi ensimmäisen SQL-standardin vuonna 1986. MySQL-tietokanta näki päivänvalon vuonna 1995 ja se on saatavissa GNU /GPL-lisenssillä tai kaupallisella lisenssillä.

2.2 Ohjelman tietokannan rakenne

Muokatessa ohjelmaa oliopohjaiseksi tietokannan rakennetta ei muutettu millään tavalla. Se siis säilyi versioiden välillä samanlaisena.



Kuva 1 Tietokannan rakenne

Tietokanta pitää sisällään henkilötaulun, jossa sijaitsee kerätyt tiedot henkilöistä. Näitä tietoja ovat henkilöiden etunimi, sukunimi, sähköpostiosoite, puhelinnumero, katuosoite, maa, maanosa, tutkimusala. Tutkimusalaa voikuvaila kolmella määrätyllä sanalla (esimerkiksi atmosphere, chemistry, biology). Valikoima sanoista löytyy

ilmoittautumislomakkeesta. Pienemmät taulut, kuten johtoryhmä, pitävät sisällään avaimen, jolla saadaan henkilötaulusta haettua henkilöiden tietoja, jotka kuuluvat kyseessä olevaan jäsenyyteen, kuten tässä tapauksessa johtoryhmään.

Näitä jäsenyyksiä saattavat olla myös erilaiset järjestettävät kokoukset, jonkun tietyn ammattiryhmän tietty osa tai mikä tahansa ryhmä, jonka ohjelman käyttäjä on todennut tarpeelliseksi. Käyttäjä pystyy luomaan ohjelmalla uuden tietokannan ja antamaan tälle haluamansa nimen sekä lisäämään tietokantaan haluamansa henkilöt.

2.3 PHP

PHP on rekursiivinen akronyymi sanasta "PHP: Hypertext Preprocessor". PHP on tullut suosituksi skriptikieleksi, joka soveltuu erityisesti Web-sovelluskehitykseen. PHP on HTML-koodin yhteyteen sijoitettava skriptauskieli, jolla www-sivuihin saadaan kätevästi dynamiikkaa. Kielen syntaksi muistuttaa C:n, Javan ja Perlin syntaksia ja on helppo oppia. [2.]

PHP on toteutettu kaikille yleisimmille käyttöjärjestelmille. PHP:ssä on myös valmiina tuki lukuisalle joukolle tietokantapalvelimia ja Web-palvelin tekniikoita. Tämä tarkoittaa sitä, että erilaisia tietokantoja on mahdollisuus käsitellä todella helposti.

PHP on palvelimella suoritettava skriptauskieli, jolla saadaan www-sivuille monimutkaistakin dynamiikkaa varsin pienellä vaivalla. Yksi PHP:n monista hyvistä puolista on, että koodi sijoitetaan suoraan HTML-koodin yhteyteen.

PHP tukee myös olio-ohjelmointia versiosta 4 alkaen. Luokkien määrittely onnistuu kuten missä tahansa yleisesti käytetyssä olio-ohjelmointikielessä. Luokkien määrittelyn jälkeen niitä voidaan käyttää PHP-koodissa. Olion luomiseksi on ensin määriteltävä malli, josta se voidaan luoda. Tämä malli on luokka. Luokka esitellään class-avainsanalla kuten Javassakin. Kun luokka on luotu, sitä käytetään pohjana, josta voidaan luoda tarvittava määrä olioita luokan ilmentymiksi. Olio (object) luodaan new-lauseella.

PHP:n versiossa 5 on uusittu/paranneltu olio-ohjelmointimallia, jossa on parantunut suorituskyky. Tämä johtuu siitä, että PHP käsittelee oliota viittauksilla, joten koko oliota ei enää tarvitse kopioida, kun se lähetetään muuttujalle tai metodille.

Aikaisemmin ei PHP:ssa ole luokalla ollut kuin yksi mahdollinen näkyvyysaste (julkinen, public). PHP5:ssä luokka voi näkyä joko julkisesti (public) tai yksityisesti (private) vain olion muille metodeille. Luokan metodit eivät kuitenkaan voi käyttää isäntäluokkansa yksityisiä attribuutteja tai metodeita, joten tätä varten on kehitetty erillinen protected-määre. Tätä määrettä käytettäessä yksityinen ominaisuus tai toiminto näkyy myös luokan periville luokille.

Näkyvyysasteiden avulla voidaan erotella toisistaan luokan sisäiset toiminnot ja sen käyttöliittymä. Luokkaa käyttävän ei tarvitse tietää sen sisäisestä toiminnasta mitään, riittää vain kontrollien hallitseminen. Jotta käyttö olisi mahdollisimman helppoa ja vähemmän altista virheille, kannattaa metodit ja attribuutit pitää aina vain välttämättömällä näkyvyysasteella. Jos määrettä ei anna, tulee metodeista tai attribuutista aina julkinen. Tämä johtuu pitkälti siitä syystä, että saataisiin yhteensopivuus PHP4:n kanssa.

PHP5 sisältää tuen myös abstrakteille luokille ja metodeille. Jos olio sisältää abstrakteja metodeja, tulee sen myös olla abstrakti luokka. Abstraktista luokasta ei voida sellaisenaan luoda oliota, vaan siinä määritellään ne attribuutit ja metodit, jotka sen perivällä luokalla on. Lisäksi voidaan määritellä abstrakteja metodeja, joille ei anneta sisältöä vielä abstraktissa luokassa, mutta sen perivässä luokassa kyseinen metodi tulee olla. Abstrakti luokka määritellään varatulla sanalla *abstract*, samoin kuin siinä olevat abstraktit metoditkin.

PHP5 on ottanut käyttöön myös rajapintaluokat. PHP:ssä luokka voi periä vain yhden luokan. Tosimaailman tarpeisiin kuuluu kuitenkin, että yksi asia voi kuulua useampaan eri joukkoon. Tällöin on mahdollista käyttää rajapintoja. Rajapinta voi sisältää vain julkisia metodeja, joita ei ole määritelty. Täten rajapintaa käyttävän luokan tulee määrittää kaikki nämä metodit.

Rajapinta tarjoaa siis käyttäjälle varman tiedon siitä, mitä metodeja luokan käyttöliittymä ainakin tarjoaa. Samaa rajapintaa käyttävät eri luokistakin luodut oliot voidaan näin käsitellä vaivattomammin samoilla toimenpiteillä. Rajapinta esitellään pitkälti samaan tapaan kuin luokkakin, sanan '*class*' tilalla käytetään vain sanaa '*interface*'. Sanalla '*implements*' puolestaan ilmaistaan sellaisia määrättyjä pilkulla erotettuja rajapintoja (mikäli niitä on useampia), jotka esiintyvät luokan esittelyssä mahdollisen *extends*-määritteen jälkeen.

PHP5:n uusia ominaisuuksia ovat myös konstruktorit ja destruktorit. Aikaisemmin PHP:ssä konstruktori oli toteutettu luokan nimen metodilla. Uusi tapa on kutsua `__construct()` metodia. PHP ei ole ennen versiota viisi tukenut luokan tuhoamista (*destructor*)-metodin avulla. PHP 5 käynnistää

`__destruct()` -metodin, mikäli viimeinen viittaus olioon tuhoetaan. Luokan tuhoaminen vapauttaa muistin ja suorittaa muut metodille määritellyt tehtävät.

Lisäksi viidenteen versioon on tullut paranneltu MySQL-tuki, XML-tuki ja Web services -tuki sekä lisätty poikkeuskäsittelijä olioille. Myös Zend engine on päivittynyt versioon 2. Suurin osa neljännen version koodista pitäisi toimia myös versiossa 5, mutta kuitenkin viidenteen versioon on tullut esimerkiksi uusia varattuja sanoja, kuten esimerkiksi oliopuolelta interface ja throw [3.]

PHP5 on kehittänyt valtavasti PHP:n olio-ohjelmointiominaisuuksia. Ohjelmointi tapahtuu luomalla uusia olioluokkia tai laajentamalla vanhoja. Aivan kuten Javassakin. Tällä toteamalla voisi sanoa, että PHP on aito oliokieli. PHP:sta kuitenkin puuttuu esimerkiksi säikeiden käsittely ja automaattinen roskien keruu, joka on hoidettu Javassa erinomaisesti. Sen sijaan poikkeusten käsittely on mahdollista ja virheiden käsittelykin sujuu varsin tyylikkäästi.

2.4 Olio-ohjelmointi

Olio-ohjelmoinnin tärkeimpiä käsitteitä ovat luokka, periytyminen ja monikäyttöisyys. Luokka on klassisen oliomallin peruskäsite, joka vastaa lähinnä perinteisen ohjelmointikielen tietotyyppiä. Luokka määrittelee olion,

luokasta luotavan ilmentymän, ominaisuudet. Luokan käyttäjää kutsutaan asiakkaaksi.

Periytyminen mahdollistaa yliluokan ominaisuuksien käyttämisen aliluokassa. Luokat ja niiden väliset periytymissuhteet muodostavat luokkahierarkian. Ydinajatuksena oliopohjaisessa ohjelmassa on liittää tiedot ja niitä käyttävät rutiinit yhteen olioön, jolla on tunnusomaisia piirteitä ja tietynlainen käyttäytymismalli.

Koska PHP on OOP (Object-Oriented Programming), siitä on hyvä ottaa kaikki "tehot" irti. PHP:ssa on mahdollisuus luoda omia luokkia samaan tapaan kuin esimerkiksi Javassa. Luokkamäärittely tapahtuu kirjoittamalla class. Oliopohjaisella ohjelmoinnilla saadaan koodiin selkeyttä, uudelleen käytettävyyttä, siirrettävyyttä ja helpottaa ohjelman ylläpitoa. Jos ohjelmaan halutaan tulevaisuudessa lisätä uusia toimintoja, se tapahtuu helposti liittämällä ohjelmaan uusia olioita, jotka hoitavat uudet toiminnot.

Tällä hetkellä ohjelmassa on esimerkiksi olioilla tuotettu tietokantayhteys. Hankalinta olio-ohjelmointiin siirtymisessä on suunnitteluvaihe, jossa työmäärä kasvaa huomattavasti. Kaikki ongelmat on pilkottava pienempiin osiin, joista jokainen osa hoitaa vain omaa osaansa ongelman ratkaisusta. Kun ollaan löydetty varsinainen olio, on ymmärrettävä, voidaanko kyseinen olio toteuttaa perimällä se toisesta oliosta vai koostamalla se useista itsenäisesti toimivista olioista [5.]

2.5 Siirtyminen funktioista olioihin

Perinteinen funktio on joukko komentoja, jotka suoritetaan peräjäälkeen. Tämän ajattelutavan mukaan varsinaiset käsiteltävät tiedot eivät liity

mitenkään ohjelman rakenteeseen, vaan ohjelmoinnin vaikeus onkin ollut kokonaisuuksien hallinnassa: mitkä funktiot kutsuvat mitäkin funktioita ja mitä tietoa funktiot muuttavat.

Nykyinen tietokoneiden käyttötapana on muuttanut tietokoneohjelmien perusrakennetta: ohjelmat ovat nykyään voimakkaasti vuorovaikutteisia, kuten tässäkin tapauksessa. Tapahtumapohjaisuus tarkoittaa sitä, että käyttäjä saa toiminnoillaan aikaan tapahtumia, joihin ohjelma reagoi. Oliopohjaisen ohjelmoinnin ydinajatus on liittää tiedot ja niitä käyttävät rutiinit eli funktiot yhteen olioon, jolla on tunnusomaisia piirteitä ja tietynlainen käyttäytymismalli [4].

Käyttöliittymän rakentaminen oliopohjaiseksi oli siis helppo aloittaa miettimällä, mitkä funktiot soveltuisivat saman luokan alle. Tämän jälkeen täytyi miettiä käyttöliittymän pääohjelman uudelleen rakentamista siten, että kutsuttaisiin vanhoja funktioita olioiden kautta eikä suoraan funktiota. Tässä tapauksessa oikaistiin hieman ja tehtiin osista moduuleita joihin koottin yhteensopivat funktiot ja tulevat luokat. Tarkemmin tehdyt muutokset selitetään luvussa 3.

3 TEKNINEN MUUTOS

Ohjelman varsinainen koodillinen muutos tapahtui joidenkin funktioiden muuttamisella olioiksi ja kaikkien funktioiden ja olioiden järjestäminen yhtenäisiksi moduuleiksi.

3.1 Alkuperäinen versio

Alkuperäinen ohjelma koostui seuraavista moduuleista, joita on paranneltu korjattuun versioon.

- addedit
- UI
- search
- arrays
- home
- membership
- printer.

Funktioiden toiminta on seuraavanlaista.

UI.php-moduuliin on koottu sivuston ulkoasun ja siihen kuuluvan päävalikon ja sen toimintojen käsittely. Ohjelma hakee halutun kohdan switch-case-rakenteen avulla ja jokaisessa kohdassa kutsuu haluttua funktiota, joka

löytyy jostain muusta moduulista. UI.php hoitaa myös yhteyden ottamisen ja sen ylläpitämisen tietokantaan.

Arrays-moduuli pitää sisällään kaikki ohjelman taulukot. Taulukkoja on seitsemän kappaletta ja ne pitävät sisällään erilaista muuttumatonta tietoa. Taulukot pitävät sisällään ohjelman valikon, maat, maanosat, tiedot työstä kolmelle eri hakusanalle (kolme eri taulukkoa) ja lomakkeella kerättävät tiedot. Tällä tavoin ohjelma löytää valikoihin tarvittavat tiedot suoraan hakemalla ne taulukoista. Tietoja on helppo lisätä tauluun, jolloin koko ohjelma saa samanaikaisesti yhteiset tiedot.

Addedit-moduuli pitää sisällään funktiot add ja edit sekä add_edit_form. Funktio add_edit_form luo halutunlaisen kaavakkeen, joka on joko tyhjä tai täynnä tietoa riippuen siitä, mitä funktiota on kutsuttu. Kutsuttaessa funktiota add-ohjelma luo tyhjän lomakkeen, johon voi täyttellä haluamansa tiedot ja siirtää ne tietokantaan painamalla add-nappia. Funktio edit luo lomakkeen, jossa on kaikki tiedot halutusta henkilöstä. Tietoja pystyy päivittämään lomakkeelle suoraan, jonka jälkeen painetaan update-nappia. Tämän jälkeen ohjelma päivittää tiedot tietokantaan. Edit-funktiota pääsee käyttämään vain, jos on valittu henkilö, jonka tietoja haluaa muokata. Tällaiseen tilaan pääsee esimerkiksi search-funktion avulla, jossa henkilöiden tietojen perässä on update-linkki tietojen päivitystä varten.

Search-moduuli pitää sisällään vain hakufunktion. Funktio luo ensin lomakkeen, jolle voidaan syöttää halutun henkilön/henkilöiden nimiä, tutkimusala, maata tai maanosaa. Annetuista hakuehdoista riippuu, montako haluttua nimeä löytyy. Voidaan esimerkiksi hakea kaikki Euroopan kemistit tai vaikka kaikki Aasin meribiologit ja niin edelleen. Kun tiedot on annettu, painetaan search-nappia, jolloin ohjelma tekee haun käyttäen

search-funktion hakutoimintoa. Tämän jälkeen samainen funktio syöttää tiedot ruudulle käyttäjän katseltavaksi. Tietojen perään on laitettu mahdollisuus poistaa henkilö tietokannasta tai vaihtoehtoisesti päivittää käyttäjän tiedot.

Membership-moduulista löytyvät funktiot `mysql_table_exist`, `add_table`, `drop_table`, `add_from_table`, `drop_from_table`, `show_from_table` ja `show_selected_table`. `mysql_table_exist` tarkistaa, onko kyseistä taulua jo olemassa, jotta ei tehtäisi kahta samannimistä taulua tietokantaan. `Add_table` lisää taulun tietokantaan. `Drop_table` poistaa kokonaisen taulun tietokannasta. Tähän on kuitenkin laitettu esto, että päätaulua, jossa on kaikki tiedot ei voi tuhota. `Add_from_table` ja `drop_from_table` mahdollistavat tietojen lisäyksen ja poiston tietokannan tauluista.

Printer-moduuli pitää sisällään funktion `printer_friendly`, joka hakee search-funktion avulla tiedot ja luo niistä tekstitiedoston palvelimelle, johon tallennetaan käyttäjän haluamat tiedot.

Home-moduuli pitää sisällään funktioihin laitettut varoitus- ja ohjelman käyttöohjetiedot. Esimerkiksi ohjelman etusivun ohjeet löytyvät tästä tiedostosta.

3.2 Päivitetty versio

Päivitetyssä versiossa edellisessä osiossa mainituissa funktioista osasta on tehty luokkia ja osa on pidetty ennallaan. Lisäksi funktiot ja luokat on koottu moduuleiksi, jotka pitävät sisällään niin luokat kuin funktiotkin. Esimerkiksi `index.php` pitää sisällään tietokannan yhteyden ottamisen.

```

class main {
  function cdatabase(){
    mysql_connc(palvelin,käyttäjä,salasana)
    or die("Tietokantayhteys ei toimi, kokeile myöhemmin!");
    mysql_select_db("tietokanta");
  }

  function xxxxx(){
  }
}

```

Tätä voidaan käyttää esimerkiksi näin.

```

$connect = new main();
$connect -> cdatabase();

```

Esimerkki 1. class

Muokattu ohjelma koostui samoista moduuleista, kuin vanha versio, mutta moduuleihin on nyt koottu sekä luokkamäärittelyt ja funktiot. Seuraavassa moduulit käydään läpi muutosten mukaan. Jos muutoksia ei ole, niin funktiot toimivat samoin kuin edellisessä versiossa.

UI-moduuli pitää edelleen huolta sivuston ulkoasusta ja yhteyden ottamisesta tietokantaan. Moduuliin on luotu luokka pääohjelma, jossa sijaitsevat metodit tietokantaan, toiminto ja nayta_kayttajalle. Tietokantaan metodi hoitaa yhteydenpidon tietokannan kanssa, toiminto vastaa käyttäjän pyyntöön, että mitä tehdään. Nayta_kayttajalle huolehtii ohjelman ulkoasusta.

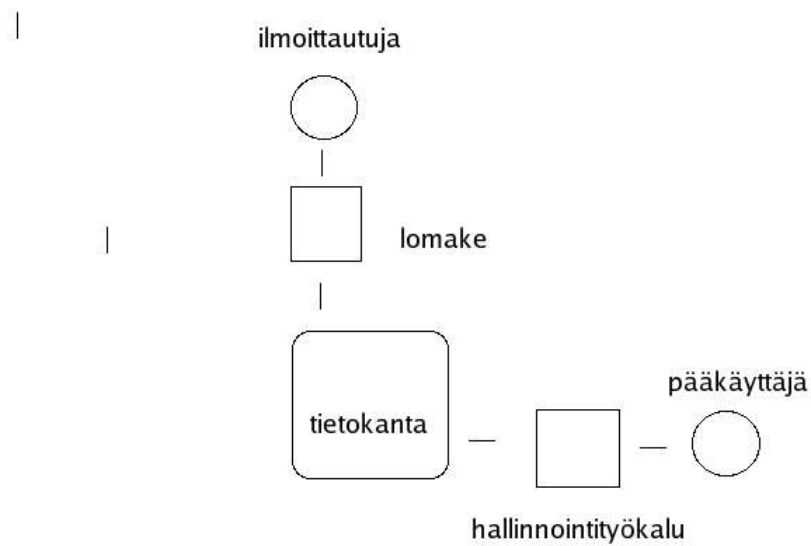
Vaikka PHP5:ssä olisi ollut mahdollisuus käyttää metodeille eri näkyvyysmääreitä (jota ei PHP4 versiossa ollut), niitä ei kuitenkaan määriteltä. Näin ollen ne ovat kaikki julkisia (public) kuten PHP:ssä vakiona on. [6] Sama pätee seuraavaan printer-moduulissa olevaan metodiin. Jos kapselointina pidetään datan ja käyttäytymisen kokoamisena yhteen yksikköön, niin siinä on onnistuttu jotenkin tässä luokassa. Jos kuvioiohin lisätään vielä tiedon piilotus, niin siinä ei varsinaisesti ole onnistuttu, sillä luokka ja sen metodit ovat julkisia.

Printer-moduuli pitää sisällään luokan printer, jolla on metodi `printer_friendly`, joka hakee Search tiedostossa olevan search funktion avulla tiedot ja luo niistä tekstitiedoston palvelimelle, johon tallennetaan käyttäjän haluamat tiedot.

Moduulit `addedit`, `search`, `arrays`, `home` ja `membership` pysyivät ennallaan. Näistä kuitenkin `home`- ja `arrays`-moduulit eivät pidä sisällään ohjelman rakenteen kannalta tärkeitä toimintoja, vaan lähinnä tietoja ja ohjelman käyttämiseen tarvittavia ohjeita.

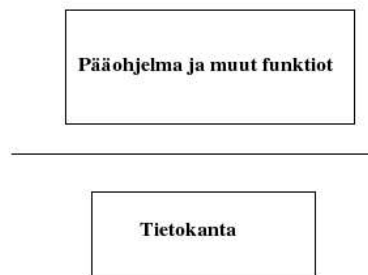
3.3 Sovelluksen rakenne

Toimintaperiaatteeltaan ohjelma on seuraavanlainen.



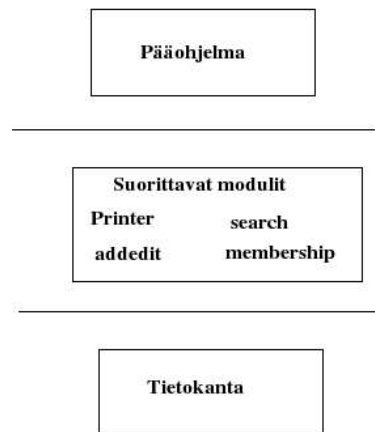
Kuva 2 Sovelluksen toimintaperiaate.

Ilmoittautuja antaa siis tietonsa lomakkeen kautta, joka tallettaa tiedot suoraan tietokantaan odottamaan hyväksymistä tai hylkäämistä. Pääkäyttäjä voi hallinnointityökalua käyttämällä poistaa tai lisätä tietokannasta ihmisiä sekä esimerkiksi lähettää heille sähköpostia.



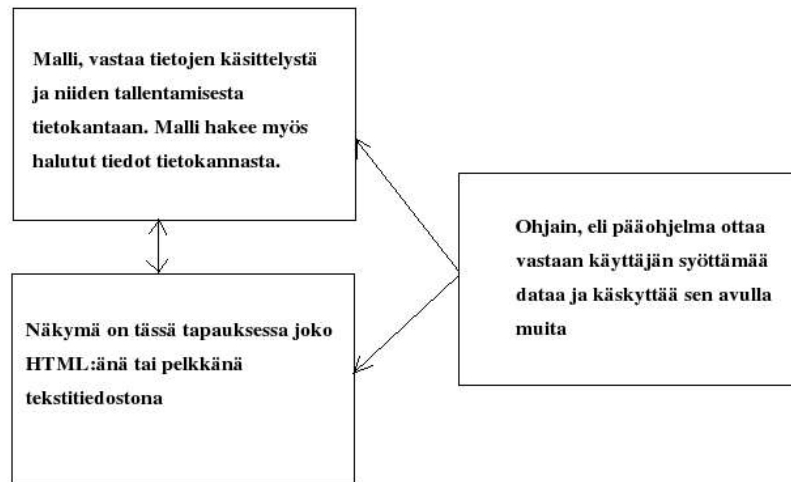
Kuva 3 Ohjelman alkuperäinen rakenne

Alkuperäisessä versiossa oli rakenteellisesti kaksi kerrosta. Oli tietokantakerros, joka hoiti tietokantayhteyden sekä tietokannan. Pääohjelma ja muut ohjelman osat olivat sulautettu seuraavaan kerrokseen. Tämä kerros hoiti kommunikoinnin käyttäjän kanssa, mahdolliset tiedonhaut tietokantakerrokselle ja näytti käyttäjälle halutut tiedot. Lisäksi tähän samaan pakettiin oli lisätty ilmoittautumislomake, joka siis toimi yhtenä osana Add/Edit-toimintoa. Uudessa versiossa tämä toiminto oli erotettu niin sanotusti omaksi ohjelman osaksi. Tälle osalle pystyi itsenäisesti antamaan tietoa, joka tallentuu tietokantaan tietyin varauksin.



Kuva 4 Ohjelman rakenne.

Uudessa versiossa (kuva 4) siirryttiin rakenteellisesti kolmeen kerrokseen. Tietokantakerros hoitaa vielä tiedon tallennuksen ja tietokannan hoitamisen. Suorittavat moduulit kommunikoivat pääohjelman ja tietokannan kanssa. Pääohjelma antaa suorittaville moduleille käskyjä, jonka mukaan ohjelma toimii. Eli esimerkiksi hae tiedot henkilöstä x. Tällöin suorittavat moduulit luovat haun tietokantaan ja näyttävät pääohjelman kautta käyttäjälle halutut tiedot henkilöstä x.



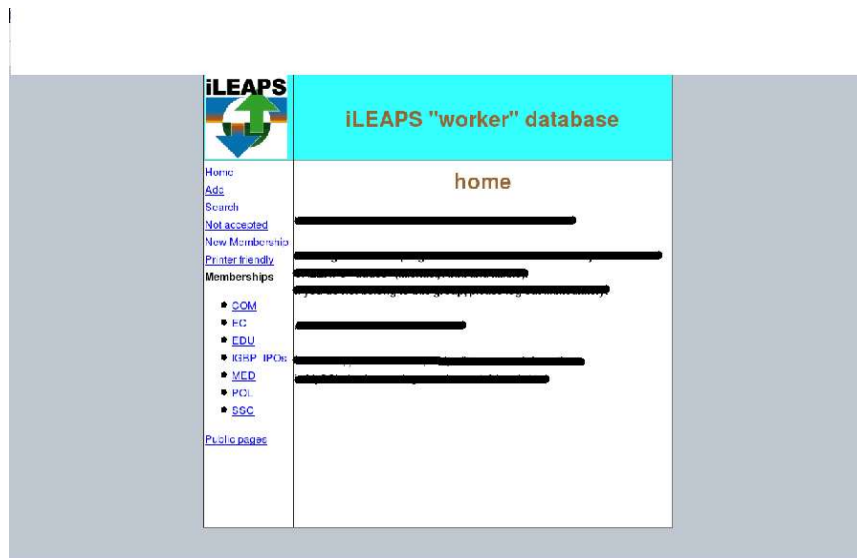
Kuva 5 MVC-malli.

MVC-mallin avulla saadaan selvitettyksi myös edellistä kerrosajattelua. Pääohjelmalla on tässä ajattelumallissa kaksijakoinen osa. Toisaalta se toimii näkymänä, jonka se antaa käyttäjälle, kun tämä antaa haluamansa komennon. Lisäksi pääohjelma kommunikoi ohjainkerroksessa muiden ”suorittavien”-moduulien kanssa, jotka varsinaisesti tekevät halutut toimenpiteet. Malliosa taas vastaa täydellisesti tietokantaosaa ja vastaa näin tiedon tallennuksesta sekä käsittelystä tietokannassa.

4 KÄYTTÄJÄN NÄKEMÄT ASIAT

Tätä insinööriötä varten sovelluksen käyttöliittymän ulkoasua ja käytettävyyttä parannettiin. Vanhasta ohjelmasta ei valitettavasti löytynyt enää kuvia, joten ulkoasun vertailu on jokseenkin mahdotonta. Ohjelma on kuitenkin aktiivisessa käytössä, joten ainakin käytettävyyttä onnistuttiin parantamaan. Ohjelmaa on nyt entistä helpompi käyttää.

4.1 Hallinnointityökalu



Kuva 6. Ohjelman etusivu

Pääsivulle tultaessa (kuva 6) vasemmalla puolella on valikko, josta voidaan valita haluttu toiminto. Mahdollisia toimintoja ovat Home, Add, Search, Not accepted, New membership, Printer friendly ja Public pages. Home-nappia painettaessa päästään takaisin aloitussivulle. Lisäksi sivulta löytyy ohjeita sovelluksen käyttöä varten ja kehoitus poistua sivustolta, jos oikeuksia ei

oikeasti ole eikä sovellusta osata käyttää. Tällä yritetään välttää mahdollisten henkilöiden ohjelman käyttö.

Add/Lisää

Kyseinen toiminto mahdollistaa yksittäisen henkilön kaikkien tietojen lisäämisen tietokantaan ja tämän hyväksymisen ilman, että tämä joutuu odotuslistalle (Not accepted).

Teknisesti tämä on toteutettu seuraavalla tavalla. Add_edit-luokka luo tietojen kyselylomakkeen, joka tarkistaa ensin onko, kyseessä tietojen lisäys vai päivitys. Jos kyseessä olisi päivitys, halutut tiedot haettaisiin ensin tietokannasta lomakkeelle. Kun kyseessä on tietojen lisäys, ohjelma luo tyhjän lomakkeen, jolle tietoja voidaan syöttää.

Kun lomakkeen tiedot on täytetty/päivitetty, ne siirretään tietokantaan talteen. Lomakkeen tietojen avainsanat on jaettu kolmeen ryhmään, joilla pyritään luokittelemaan ihmiset oikeisiin tieteenaloihin löytyvät valmiista taulukoista. Jos mahdollisia uusia avainsanoja ilmenee nyt tai tulevaisuudessa, ne pystytään helposti lisäämään suoraan taulukoihin. Itse ohjelmakoodia ei siis tarvitse muuttaa.

Search/Haku

Tämä toiminto mahdollistaa yksittäisen henkilön tai henkilöryhmän etsimisen nimen, maan, maanosan tai avainsanojen avulla. Haetun henkilön/henkilöiden tiedot voidaan poistaa tai tietoja voidaan päivittää kyseessä olevan henkilön tietojen takana olevilla delete- (poista) ja update- (päivitä) linkeillä.

iLEAPS **iLEAPS "worker" database**

search

[Home](#)
[Ads](#)
[Search](#)
[Not accepted](#)
[New Memberships](#)
[Printer friendly](#)
[Memberships](#)
 • [COM](#)
 • [EC](#)
 • [EDU](#)
 • [GHP](#) [IPCs](#)
 • [MED](#)
 • [PCL](#)
 • [SSC](#)
[Public pages](#)

Name:
Country:
Continent:
Keyword 1:
Keyword 2:
Keyword 3:
All: ☐

Kuva 7. Ohjelman hakutoiminto

Tämä on toteutettu siten, että hakumoduli luo ensin hakulomakkeen, jonne voidaan syöttää tarvittavat tiedot, esimerkiksi nimi tai maa. Tämän jälkeen tiedot annetaan funktiolle `search_result`, joka toteuttaa haun ja syöttää ruudulle tuloksen (koodi löytyy liitteestä 1).

Not accepted/Ei hyväksytty

Kyseisellä toiminnolla näkee, keitä ilmoittautuneita ei ole vielä hyväksytty varsinaiseen tietokantaan. Tällä toiminnolla voidaan poistaa mahdolliset häirikköyrytykset, kuten Aku Ankat ja Mikki Hiiret, ennen kuin ne pääsevät varsinaisille postituslistoille näkyviin. Jos henkilöä ei ole hyväksytty, tietoja ei saada näkyviin hakutoiminnolla eikä liitettyä valmiina oleviin ryhmiin. Jos tiedoista löytyy esimerkiksi Mikki Hiiri, se voidaan tuhota tietokannasta kokonaan. Tietojen haku henkilöistä, joita ei ole vielä hyväksytty, on toteutettu seuraavalla tavalla. Koodi katso liite 2.

New membership/uusi jäsenyys

Tällä toiminnolla voidaan luoda niin sanottuja uusia jäsenyyksiä mahdollisia uusia projekteja varten, jolloin on helppo nähdä, kuka on missäkin projektissa ja mitä hän siinä tekee. Kaikki "avoinna" olevat projektit näkee membership-valikon alta.

Printer friendly version/Tulostus versio

Tällä toiminnolla voidaan valita ne tiedot, joita halutaan tulostaa, esimerkiksi tiettyjen henkilöiden nimet ja arvonimet, jos halutaan esimerkiksi laittaa istumislaput kokoukseen. Toiminto luo haluttuun hakemistoon tekstitiedoston, jota päästään lukemaan ja tulostamaan suoraan selaimen avulla. Tiedosto voidaan myös tallentaa omalle koneelle ja avata vaikka office-sovelluksella. (Word, Excel).

Teknisesti tämä on toteutettu siten, että ohjelma käy hakemassa tietokannasta halutut tiedot, jonka jälkeen se tulostaa halutut tiedot tekstitiedostoon www-palvelimelle (sisäisille sivuille) tiettyyn paikkaan, josta ne voi käydä hakemassa ohjelman antaman linkin kautta.

Public pages/Julkiset sivut

Tällä linkillä voidaan kirjautua ulos ohjelmasta ja päädytään iLEAPSin julkisille sivuille.

4.2 Ilmoittautumislomake

[illegible]

Kuva 8 Julkisilla sivuilla oleva ilmoittautumislomake.

Ilmoittautumislomakkeella (kuva 8) voidaan ilmoittaa iLEAPSin henkilökunnalle omat tietonsa. Tiedot talletetaan tietokantaan, mutta ohjelma ei vielä näytä annettuja tietoja, sillä tiedot tarvitsee ensin hyväksyä pääohjelman avulla, jotta esimerkiksi hakutoiminto näyttäisi sillä hetkellä

kyseessä olevan henkilön tiedot. Tällä pyritään välttämään mahdollisten mikkihiirien ja akuankkojen joutuminen paikkoihin, joissa heidän ei kuuluisi olla.

5 PÄIVITYKSEN HYVÄT JA HUONOT PIIRTEET

5.1 Ohjelma

Päivityksellä saatiin aikaan ohjelmalle siedettävämpi ulkoasu. Myös ohjelman käytettävyys parani. Tämä näkyi eritoten tietokantahakujen sekä tietokannan ylläpidon helpottumisena. Tietokannasta pystyi tekemään monipuolisempia ja tarkempia hakuja. Tämä helpotti siis erityisesti pääkäyttäjän toimimista ohjelman kanssa.

Ohjelmaa voisi vielä parannella esimerkiksi lisäämällä siihen mahdollisuuden etsiä joistakin tietyistä tietokannoista ilmoittautuneiden henkilöiden erilaisia tieteellisiä julkaisuja tai muita vastaavia tietoja. Lisäksi ilmoittautuneiden henkilöiden olisi hyvä voida ladata erilaisia ”salaisia” tiedostoja tai laittaa niitä muiden luettavaksi. Tämä olisi siis mahdollista vain rekisteröityneillä käyttäjillä. Lisäksi ohjelmaan voisi lisätä jonkinlaisen keskustelufoorumin, jotta välttyttäisiin sähköpostitulvilta.

5.2 Moduleihin siirtymisen hyvät ja huonot piirteet

Yleisesti ottaen ohjelmistojen kehityksessä pyritään mahdollisimman tehokkaisiin, monipuolisiin, luotettaviin ja helppokäyttöisiin ohjelmiin. Laajamittaisessa työssä nämä ominaisuudet on yleensä hankala saavuttaa. Isoissa ohjelmistoissa olio-ohjelmointi on siis perusteltua. Isommat ongelmat voidaan jakaa yhä pienempiin osiin. Kun ongelma on saatu jaettua niin pieneen osaan, että se vastaa kysymykseen ”mikä”, ollaan löydetty olio ja sitä voidaan alkaa toteuttaa halutulla ohjelmointikielellä.

Tästä seuraa se, että käytettäessä olio-ohjelmointia suunnitteluvaihe saattaa olla yllättävänkin työläs ja pitkä. Jos ohjelman halutaan tulostavan muutama rivi tekstiä, sekä sen jälkeen lopettavan toimintansa, ei välttämättä ole perusteltua käyttää olio-ohjelmointia.

Tämä ohjelma on kuitenkin koodillisesti suhteellisen pieni. Vastaavan hyödyn saavuttaminen olisi siis vaatinut koodillisesti paljon suurempaa ohjelmaa. Käytännössä se aika, joka kului mahdollisten olioiden miettimiseen, olisi voitu käyttää alkuperäisen ohjelmakoodin selkeyttämiseen ihan proseduraalisessa ajattelumaailmassa. Tämän takia ohjelma koottiin moduuleihin, joissa yhdistyvät niin oliot kuin ”pelkät” funktiotkin.

Kuitenkin jos katsotaan tulevaisuuteen ja huomataan, että ohjelma on muokattu osittain oliopohjaiseksi, saavutetaan etu, joka tapahtuu ohjelman päivittämisessä. Mahdolliset uudistukset vaativat siis vain jonkun uuden olion lisäämistä ohjelmakoodiin tai vanhan muokkaamista halutulla tavalla. Tämä nopeuttaa, selkeyttää ja ennenkaikkea helpottaa ohjelman uudistamista. Mahdollisimman suuri etu oltaisiin saavutettu, jos ohjelmasta olisi tullut täysin oliopohjainen.

Vaikka olio-ohjelmissa joudutaan käyttämään enemmän aikaa suunnitteluun ja miettimiseen, korvautuu se tulevaisuudessa päivittämisen ja ohjelman toiminnan hallinnan helpottumisella.

6 YHTEENVETO

Tämän opinnäytetyön tavoitteena oli tehdä iLEAPSille toimiva tietokanta ja tietokannan hallintaväline helpottamaan tietokannan käyttöä. Lisäksi työn tavoitteena oli muuttaa työkalu olio-pohjaiseksi sekä parantaa käyttöliittymän ulkoasua. Tavoitteena oli myös tarkastella olio-ohjelmointia PHP:lla ja tarkastella PHP:n ja MySQL:n yhteistoimintaa.

Työn tuloksena iLEAPS:ille saatiin toimiva tietokanta ja siihen työkalu, jolla pystytään hallitsemaan tietokantaa. Tämän lisäksi saatiin aikaan erillinen ilmoittautumislomake, jolla pystytään ilmoittamaan omat tiedot tietokantaan. Opinnäytetyössä tehtiin myös päivitetty versio, jossa varsinaisen hallinnointityökalun rakenne on muutettu osittain olio-pohjaiseksi. Ohjelma itsessään onnistui hyvin ja on tälläkin hetkellä käytössä iLEAPSilla.

Tätä työtä voitaisiin käyttää pohjana tai jo valmiina olevana ohjelmana helpottamaan muidenkin eri yritysten/yhteisöjen tietokantojen ylläpitoa. Tämä tapahtuisi sillä edellytyksellä, että tietokanta koskisi vastaavia tietoja, joita tämä ohjelma kerää. Muuten ohjelmaa pitäisi muokata jonkin verran. Jatkokehityksenä tietokannalle voisi ohjelmoida hallinnointityökalun esimerkiksi Javalla tai C++:lla, joka ei tarvitsisi www-selainta tietokannan hallintaan. Lisäksi ohjelmaan voisi lisätä esimerkiksi keskustelufoorumin rekisteröityneille käyttäjille sekä mahdollisuuden ladata muiden julkaisuja sekä laittaa omia julkaisujaan levitykseen.

Mitä tulee tämän opinnäytetyön kirjalliseen osuuteen, niin sen olisi voinut tehdä jo aikaisemminkin, sillä itse ohjelma on ollut valmiina muutaman vuoden. Tämä olisi ollut tärkeää sen vuoksi, että ohjelmasta esiintuodut tiedot olisivat tällöin olleet tarkempia ja ajankohtaisempia. PHP ja MySQL

ovat kuitenkin kehittyneet ohjelman valmistumisen ja opinnäytetyön kirjoittamisen aikana jonkin verran.

VIITELUETTELO

- [1] Laine Harri, Tietokantojen perusteet, *Opetusmoniste*, Tietojenkäsittelytieteen laitos. 2000.
- [2] Antony D et al., PHP Manuaali, [verkkodokumentti viitattu 3.4.2006]. Saatavissa: <http://www.php.net/manual/en/> .
- [3] PHP Manuaali, [verkkodokumentti viitattu 5.4.2006]. Saatavissa: <http://fi2.php.net/manual/en/migration5.php#migration5.changes>.
- [4] Liberty Jesse, opeta itsellesi C ++.
- [5] Peltomäki Juha, Silander Simo, Java 2 Ohjelmoinnin peruskirja. 2002.
- [6] PHP manuaali [verkkodokumentti viitattu 16.12.2006]. Saatavissa: <http://fi2.www.php.net>.